

E4
日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

07.07.00
REC'D 21 JUL 2000
WIPO PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

1999年 7月 7日

JP00/4552

出願番号
Application Number:

平成11年特許願第193542号

出願人
Applicant(s):

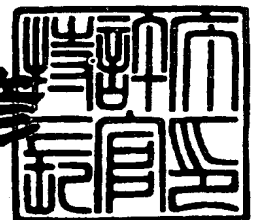
ソニー株式会社

PRIORITY
DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

2000年 5月19日

特許庁長官
Commissioner,
Patent Office

近藤 隆彦



出証番号 出証特2000-3037224

【書類名】 特許願

【整理番号】 9801153702

【提出日】 平成11年 7月 8日

【あて先】 特許庁長官殿

【国際特許分類】 H04N 1/387

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社
内

【氏名】 市岡 秀俊

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 出井 伸之

【代理人】

【識別番号】 100067736

【弁理士】

【氏名又は名称】 小池 晃

【選任した代理人】

【識別番号】 100086335

【弁理士】

【氏名又は名称】 田村 榮一

【選任した代理人】

【識別番号】 100096677

【弁理士】

【氏名又は名称】 伊賀 誠司

【手数料の表示】

【予納台帳番号】 019530

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9707387

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 信号処理方法及び装置

【特許請求の範囲】

【請求項 1】 入力信号に量子化を施して数列とし、この数列に可変長符号化を施して符号語列に変換する信号処理方法において、

上記符号語列中の一部の符号語を、長さがより短くかつ値が近似した他の符号語に置換すること

を特徴とする信号処理方法。

【請求項 2】 上記置換される符号語は、元の符号語に対してさらに量子化した場合と等価となる符号語であることを特徴とする請求項 1 記載の信号処理方法。

【請求項 3】 上記可変長符号化は、出現確率が高いほど短い符号を割り当てるエントロピー符号化であることを特徴とする請求項 1 記載の信号処理方法。

【請求項 4】 上記数列を非零の数の前に 0 が連続する個数であるランレングス (run length; run) と、上記非零の数である振幅 (amplitude; amp) との対 (run, amp) の列にて表し、元の符号語を、amp を近似した対に対応する他の符号語に置換することを特徴とする請求項 1 記載の信号処理方法。

【請求項 5】 入力されたフレームごとの画像信号に対して、上記フレームを構成する画素について、上記フレームを複数のブロックに分割し、上記ブロックに離散コサイン変換 (descrete cosine transformation; DCT) を施し、上記 DCT 変換が施された上記ブロックの DCT 係数を量子化情報に基づいて量子化し、上記量子化が施された上記 DCT 係数を 1 次元の数列に配列して上記可変長符号化を施すことを特徴とする請求項 1 記載の信号処理方法。

【請求項 6】 上記符号語の置換は、上記数列において上記 DCT 係数の高域側から順に実行し、上記数列を可変長符号化したビット量が目標値に達すると置換を終了することを特徴とする請求項 5 記載の信号処理方法。

【請求項 7】 入力信号に量子化を施して数列とし、この数列に可変長符号化を施して符号語列に変換する信号処理装置において、

上記符号語列中の一部の符号語を、長さがより短くかつ値が近似した他の符号

語に置換する手段

を有することを特徴とする信号処理装置。

【請求項 8】 上記置換される符号語は、元の符号語に対してさらに量子化した場合と等価となる符号語であることを特徴とする請求項 7 記載の信号処理装置。

【請求項 9】 上記可変長符号化は、出現確率が高いほど短い符号を割り当てるエントロピー符号化であることを特徴とする請求項 7 記載の信号処理装置。

【請求項 10】 上記数列を非零の数の前に 0 が連続する個数であるランレングス (run length; run) と、上記非零の数である振幅 (amplitude; amp) との対 (run, amp) の列にて表し、元の符号語を、amp を近似した対に対応する他の符号語に置換することを特徴とする請求項 7 記載の信号処理装置。

【請求項 11】 入力されたフレームごとの画像信号に対して、上記フレームを構成する画素について、上記フレームを複数のブロックに分割し、上記ブロックに離散コサイン変換 (descrete cosine transformation; DCT) を施し、上記 DCT 変換が施された上記ブロックの DCT 係数を量子化情報に基づいて量子化し、上記量子化が施された上記 DCT 係数を 1 次元の数列に配列して上記可変長符号化を施すことを特徴とする請求項 7 記載の信号処理装置。

【請求項 12】 上記符号語の置換は、上記数列において上記 DCT 係数の高域側から順に実行し、上記数列を可変長符号化したビット量が目標値に達すると置換を終了することを特徴とする請求項 11 記載の信号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、符号語を処理する信号処理方法及び装置に関し、詳しくは、画像信号を符号化した符号語の処理に好適な信号処理方法及び装置に関する。

【0002】

【従来の技術】

従来、静止画像、動画像などのデジタル信号の編集においては、静止画像又は動画像（以下、単に画像と称する。）の 1 画像フレーム全体を編集加工の単位と

している。特に圧縮や色空間変換等の各種変換が施されたデジタル信号に対しては、画像全体を逆変換して原信号（例えばRGB信号）を得た後に、編集加工を行って、その後再変換を施している。また、静止画像や動画等のデジタル信号は、データ量が膨大なため、一般に圧縮されて伝送又は保管されている。

【0003】

図25には、いわゆるデジタルビデオ (Digital Video; DV) の1画像フレームに対して、編集加工する場合の処理手順の例を示している。ここでは、エントロピー符号化等により符号化されている画像について行う編集加工を示している。

【0004】

編集加工は、この図25に示すように、テープ又はハードディスクから入力されたデジタル信号を、エントロピー復号化（ステップS101）、逆量子化（ステップS102）、逆離散コサイン変換 (Discrete Cosine Transform; DCT) （ステップS103）、RGB変換ブロック合成（ステップS104）の順に処理し、この処理により再生したデータに対して編集加工処理を行う（ステップS105）。

【0005】

編集したデータについては、今度は逆の変換を施す。すなわち、YCbCr変換ブロック化（ステップS106）、DCT（ステップS107）、量子化（ステップS108）、エントロピー符号化（ステップS109）の処理を順に施す。

【0006】

【発明が解決しようとする課題】

ところで、上述したように圧縮されているDVなどの圧縮画像は、この圧縮画像を構成するある単位ごとの符号量の上限が決められており、この符号量に収まるように圧縮量を制御する必要がある。

【0007】

例えば、DVでは1ビデオセグメントの符号量が所定の符号量以下に収まるように固定長化されている。このため、従来は、マクロブロック単位に量子化器番号を小さくして量子化を粗くし、それを可変長符号であるハフマン符号化するこ

とにより、符号量を制御していた。

【0008】

しかし、この方法では、必要以上に圧縮され、目標符号量を大きく下回る場合があり、結果として必要以上に画質が劣化してしまう傾向がある。例えば、医療、美術、映画をはじめとする高精細映像の制作においては、画質劣化に対する危惧から非圧縮画像が用いられる。特に、繰り返し画面合成を行う場合、劣化が累積されてしまうため、圧縮画像は一層敬遠される傾向にある。

【0009】

また、一段粗い量子化によっても目標符号量に達しない場合、目標符号量が得られるまで、量子化と可変長符号化を繰り返す必要があり、負荷の大きな処理となっていた。

【0010】

もし、これら伸長及び再圧縮に伴う劣化の問題を改善できれば、いわゆるDVなどの圧縮画像による編集がより広く受け入れられるようになると期待できる。

【0011】

本発明は、上述の課題に鑑みて提案されるものであって、圧縮画像の符号語を処理する際に画質を維持すると共に、処理の負担を低減するような信号処理方法及び装置を提供することを目的とする。

【0012】

【課題を解決するための手段】

上述の課題を解決するために、本発明に係る信号処理方法は、入力信号に量子化を施して数列とし、この数列に可変長符号化を施して符号語列に変換する信号処理方法において、上記符号語列中の一部の符号語を、長さがより短かつ値が近似した他の符号語に置換するものである。

【0013】

本発明に係る信号処理装置は、入力信号に量子化を施して数列とし、この数列に可変長符号化を施して符号語列に変換する信号処理装置において、上記符号語列中の一部の符号語を、長さがより短かつ値が近似した他の符号語に置換する手段を有するものである。

【0014】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を参照して説明する。本発明の実施の形態は、本発明をいわゆるデジタルビデオ (digital video; DV) を編集する編集装置に適用したものである。

【0015】

本実施の形態の編集装置は、いわゆるDVを対象とする合成加工処理に対して、圧縮信号を可能な限りそのまま複製することで劣化0を目指し、再量子化が必要に待った場合のみ、可変長符号語単位で圧縮を行うものである。

【0016】

編集装置は、図1に示すように、画像を撮像して画像信号を出力するカメラデッキ11と、画像データを蓄積するハードディスクドライブ12とを有している。

【0017】

カメラデッキ11は、画像を撮像して画像信号を出力すると共に、この画像信号をいわゆるDV規格によって記録されるテープ状媒体、いわゆるDVテープに記録するものである。また、カメラデッキ11は、いわゆるDVテープに記録された画像データを再生して画像信号を出力する。

【0018】

ハードディスクドライブ12は、画像データや音声データのいわゆるAV (audiovisual) データを蓄積する大容量の記録メディアである。ハードディスクドライブ12は、カメラデッキ11との間で、例えばいわゆるIEEE1394規格に従って画像データを送受する。

【0019】

編集装置は、制御状態を表示する制御モニタ13と、編集のために画像を表示する画像モニタ14とを有している。

【0020】

制御モニタ13は、編集装置における画像処理などの制御状態をユーザに表示するものである。制御モニタ13としては、例えば文字や記号を表示することが

可能な陰極線管 (cathode-ray tube; CRT) を用いることができる。

【0021】

画像モニタ14は、編集装置で編集中の画像や、編集装置で編集された画像をユーザに表示する。画像モニタ14としては、例えば高精細のCRTを用いることができる。

【0022】

編集装置は、文字データや記号データが入力されるキーボード15と、位置情報を入力されるマウス16とを有している。

【0023】

キーボード15は、文字や記号に対応する複数のキーにより、文字データや記号データが入力される。マウス16は、滑動されたり、そのボタンをクリックされることにより、位置情報等を入力される。

【0024】

編集装置は、画像データの編集その他の処理を行うパーソナルコンピュータ17を有している。パーソナルコンピュータ17は、CPU、ROM、RAM等を有し、例えばRAMに読み込まれた処理プログラムに基づいて、画像編集などの処理を実行する自立型の情報処理装置である。

【0025】

パーソナルコンピュータ17は、例えば内蔵ハードディスクドライブのような記憶装置を内蔵している。この記憶装置には、画像を編集するための画像編集ソフトウェア20が記録されている。

【0026】

すなわち、パーソナルコンピュータ17は、画像編集ソフトウェア20として、グラフィックユーザインターフェース (graphic user interface; GUI) 21と、複数の画像を合成する合成処理プログラム22と、一旦は伸長した圧縮画像データを再び圧縮する再圧縮処理プログラム23とを有している。

【0027】

また、画像編集ソフトウェア20は、例えばPCIボード上のLSIとしてハードウェア化され、PCIバスを介して動作するように実装することもできる。

【 0 0 2 8 】

これらの画像編集ソフトウェア 2 0 で、再圧縮処理プログラム 2 3 は、いわゆる D V 規格などに基づく可変長符号語を再圧縮する場合に、符号語を他の符号語に置換することによって、最小限の量子化を行う。すなわち、本実施の形態においては、符号語に対する再量子化を最小限にすることにより、再量子化によるデータの劣化を最小限に止めようとするものである。

【 0 0 2 9 】

この再圧縮プログラムにより、パーソナルコンピュータ 1 7 を中心とする編集装置において、画像圧縮の処理が行われる。編集装置にて行われる本実施の形態の画像圧縮の方法については、さらに後述する。

【 0 0 3 0 】

パーソナルコンピュータ 1 7 は、例えばいわゆる I E E E 1 3 9 4 規格に基づいて、カメラデッキ 1 1 との間で、画像データなどを相互に送受することができる。パーソナルコンピュータ 1 7 は、ハードディスクドライブ 1 2 との間で、例えばいわゆる U l t r a S C S I 規格に従って、画像データを高速に送受することができる。

【 0 0 3 1 】

続いて、上述した編集装置の構成を、画像編集ソフトウェア 2 0 との関連において説明する。

【 0 0 3 2 】

図 2 に示すように、編集装置には、記録媒体として、いわゆる C D やいわゆる D V D のようなディスク状媒体 3 1 や、ハードディスク 3 2 や、いわゆる D V テープのようなテープ状媒体 3 3 が備えられる。これらの記録媒体には、編集装置で編集する画像や、画像編集ソフトウェア 2 0 が記録される。

【 0 0 3 3 】

編集装置においては、例えば、ディスク状媒体 3 1 はパーソナルコンピュータ 1 7 に挿入されて用いられる。また、ハードディスク 3 2 は、ハードディスクドライブ 1 2 やパーソナルコンピュータに備えられる。テープ状媒体 3 3 は、カメラデッキ 1 1 にて用いられる。

【 0 0 3 4 】

編集装置においては、ディスク状媒体 3 1 は第 1 のインターフェース 3 4 を、ハードディスクドライブ 3 2 は第 2 のインターフェース 3 5 を、テープ状記録媒体 3 6 は第 3 のインターフェース 3 6 を、それぞれ介してデータを送受している。第 1 のインターフェース 3 4、第 2 のインターフェース 3 5 及び第 3 のインターフェース 3 6 は、記録媒体と送受するデータの形式やタイミングを調整する。

【 0 0 3 5 】

編集装置には、ユーザから制御データを入力されるユーザ制御装置 3 7 と、文字や画像を表示する表示装置 3 8 とを有している。

【 0 0 3 6 】

ユーザ制御装置 3 7 には、編集装置にて実行される画像編集ソフトウェア 2 0 に対するユーザによる制御が入力される。表示装置には、編集装置における画像編集ソフトウェア 2 0 の実行の状況や、この画像編集ソフトウェア 2 0 で編集している画像が表示される。

【 0 0 3 7 】

編集装置には、データを記憶するメモリ 3 9 と、高速の記憶バッファであるキャッシュメモリ 4 0 と、画像編集ソフトウェア 2 0 を実行する CPU 4 1 とが設けられている。

【 0 0 3 8 】

メモリ 3 9 には、揮発性の RAM や不揮発性の ROM が用いられ、画像データや画像編集ソフトウェア 2 0 を記憶する。キャッシュメモリ 4 0 は、CPU 4 1 と一体となってデータの高速にバッファすることにより、データ処理の速度を向上させる。

【 0 0 3 9 】

CPU 4 1 は、例えばメモリ 3 9 に記憶された画像編集ソフトウェア 2 0 に処理手順に基づいて、画像編集のためのデータ処理を実行する。CPU 4 1 で実行される画像編集ソフトウェア 2 0 は、例えばユーザ制御装置 3 7 に入力されたユーザによる制御に基づいて、その実行する処理を決定する。CPU 4 1 は、画像データを処理して結果として、画像データを表示装置 3 8 に表示したり、ディス

ク状記憶媒体 31、ハードディスク 32 又はテープ状記憶媒体 33 に記録したりする。

【0040】

続いて、編集装置における本実施の形態の画像処理の方法の前提となる、いわゆる DV 規格による圧縮画像について説明する。本実施の形態の編集装置は、いわゆる DV 規格による圧縮画像を編集することを想定している。

【0041】

図 3 には、いわゆる DV フォーマットの仕様の概要を示している。DV フォーマットにおける画像フレーム FR は、固定長のビデオセグメント VS により構成されている。

【0042】

各ビデオセグメント VS は、5 つの固定長の圧縮されたマクロブロック MB より構成されている。各ビデオセグメント VS を構成する圧縮されたマクロブロック MB は、1 画面において分散、いわゆるシャッフルされている。

【0043】

各圧縮されたマクロブロック MB は、6 つの圧縮されたブロック BL より構成されている。各圧縮されたブロック BL における有効なデータの長さは様々である。有効なデータの最後には、ブロック BL における有効なデータの終端を示す EOB (End Of Block) がマークされている。

【0044】

各圧縮されたブロック BL が格納される領域の先頭位置と大きさは固定されている。一のブロック BL に記録すべきデータが固定された領域に収まらない場合には、そのデータは他の領域に記録される。

【0045】

すなわち、固定された領域に収まらなかったデータは、その圧縮されたブロック BL を含む圧縮されたマクロブロック MB における、又はその圧縮されたマクロブロック MB を含むビデオセグメント VS における、他の圧縮されたブロック BL において、有効なデータが記録されていない余白の領域に格納される。

【0046】

例えば、第1の圧縮されたブロック BL_1 に書き込むべきデータがこの領域に収まらないことがある。この場合には、矢印 a に示すように、第2のブロック BL_2 の余白の領域に第1の圧縮されたブロック BL_1 に収まらなかったデータが書き込まれる。

【0047】

このようないわゆるDV規格のフォーマットの仕様について、図4に示すフローチャートを参照して、その構成の手順を詳細に説明する。本実施の形態における画像圧縮の方法は、このような手順により作成されたいわゆるDV規格の圧縮画像に対して適用されるものである。

【0048】

いわゆるDV規格により圧縮される画像信号は、通常は、輝度信号Y、二つの色差信号CR, CBのコンポーネント信号として供給される。このコンポーネント信号について、画像信号を構成する画像フレームFRを単位として、色差信号CR, CBは、輝度信号Yをサンプリングする周波数に比較して1/2の周波数でサンプリングされる。さらに、色差信号CR, CBは、データが1/2になるようにサブサンプリングされる。

【0049】

ステップS11においては、1画像フレームFRのデータは、2次元離散コサイン変換 (descrete cosine transformation; DCT) の基本単位である 8×8 画素のブロックBLに分割される。

【0050】

上述したように、輝度信号Yと色差信号CR, CBはサンプリング数が異なるために、輝度信号Yの画素から構成される4ブロックBL、色差信号CR, CBから構成される各1ブロックBLが画面上同じ位置の同じ面積に相当する。マクロブロックMBは、この6ブロックBL単位で構成される。

【0051】

ステップS12においては、ビデオセグメントVSを構成する5マクロブロックMBが、1画像フレームFRの画面内から所定のルールでシャッフルして集め

られる。シャッフルする目的は、ビデオセグメントVSに含まれる情報量を均一化するためである。ビデオセグメントVSは、所定のビット数を超えないように圧縮されて固定長化される。

【0052】

ステップS13においては、ビデオセグメントVS内の各ブロックBLは、被写体がカメラに対して動いているかを検出する動き検出により、静止モードか動きモードか判断される。

【0053】

ステップS14においては、ステップS11でのシャッフルにより構成された各ビデオセグメントVSは、その中の各ブロックBLごとにDCT変換される。また、高域にいくにつれて小さくなるような重み付けが施される。

【0054】

ステップS12により静止モードとされた場合には、ブロックBL内の 8×8 画素に対して 8×8 の2次元DCT変換（ $8-8DCT$ ）が施される。このDCT変換により、 8×8 のDCT係数（以下単に係数と称する）が選られる。

【0055】

ステップS12により動きモードとされた場合には、 8×8 画素から構成されるブロックBLを第1フィールドと第2フィールドの各々の 4×8 から構成されるブロックBLに分割し、各 4×8 画素に対して 4×8 の2次元DCT変換（ $4-8DCT$ ）を施す。このDCT変換により2組の 4×8 の係数が得られる。

【0056】

この動きモードにより、カメラに対して被写体が動いている場合に、このステップS14でのDCT変換の際に、インターレースによるエネルギー分散を防止することができる。

【0057】

以下のステップS15及びステップS16においては、DCT変換と重み付けされた係数に対する、量子化に関連する処理が施される。いわゆるDV規格による量子化の処理は、図5に示す量子化表に基づいて行われる。

【 0 0 5 8 】

各係数に適用される量子化のステップとなる量子化ステップは、マクロブロックMB単位に決められる「量子化器番号」(QNO)と、ブロックBL毎に決められる「クラス番号」(CNO)と、DCT変換されたブロックBL内の係数の周波数帯によって決まる「エリア番号」(ANO)とによって決定される。

【 0 0 5 9 】

以下、これの3つの番号の組み合わせを「量子化情報」と呼び、 $\langle \text{QNO}, \text{CNO}, \text{ANO} \rangle$ と記すことにする。なお、これらの量子化器番号、クラス番号及びエリア番号の意味については、以下で説明する。

【 0 0 6 0 】

ステップS14において、全てのブロックBLがDCT変換と重み付けがなされたビデオセグメントVSは、一旦バッファに格納され、データ量推定によりこのビデオセグメントを量子化する量子化器が選択される。

【 0 0 6 1 】

ステップS15においては、図5の量子化表に示したように、ビデオセグメントVS内の各ブロックBLは4種類のクラスの内1つにクラス分けされる。量子化器を構成する量子化ステップは、そのクラス番号(CNO)によって異なる。

【 0 0 6 2 】

これは、クラス番号が大きいブロックBLほど、そのブロックBLの精細度、すなわちアクティビティが大きいことを示しており、相対的に粗く量子化することを目的としている。

【 0 0 6 3 】

なお、係数の絶対値が255を超える係数を含むブロックBLは、クラス番号(CNO)が3となり、イニシャルシフトと呼ばれる操作を量子化前に受ける。イニシャルシフトでは、その係数は2で除算される。

【 0 0 6 4 】

ステップS16においては、DCT変換により得られた各ブロックBLの係数は、量子化ステップと呼ばれる整数値で除算される。この操作は、量子化又は再

量子化と呼ばれる。実際には、図5の量子化表に示したように、16種類の量子化器を予め用意し、その中から1つの量子化器を選択する。

【0065】

量子化器番号(QNO)が減少するにつれて、ビデオセグメントVSのビット数も減少していく。また、量子化器番号が減少すると各係数は粗く量子化されるため、この量子化による誤差、すなわち量子化誤差も増大し、画質劣化につながる。

【0066】

従って、ビデオセグメントVSに与えられたビット数、すなわちターゲットビット数を超えないような最大の量子化器番号(QNO)を符号量推定により選択する。符号量推定により量子化器番号が選択されると、バッファに貯えられていたビデオセグメントはこの量子化器番号により実際に量子化される。

【0067】

図5に示した量子化表によると、各量子化器はクラス番号(CNO)による4つの量子化ステップのセットで構成される。図6中のA及び同図中のBは、静止モード及び動きモードの場合のそれぞれについてのエリア分けを示している。1ブロックBL内の各係数は4つのエリアに分割されて、そのエリアによって異なる量子化ステップで量子化される。

【0068】

図6中のAに示すように、静止モードの場合には、1ブロックBLは1個のDC係数と63個のAC係数から構成される。

【0069】

図6中のBに示すように、動きモードの場合には、本来は2個の4×8画素のブロックBLのそれぞれが1個のDC係数と31個のAC係数から構成される。しかし、これらの2つのモードを画一的に扱えるようにするために、各4×8画素に対するDCT変換後、各ブロックBLの同じ次数の係数どうしで和と差を取り、8×8画素のブロックBLを構成する。

【0070】

この処理により、動きモードのブロックBLも、静止モードのブロックBLと

同様に1個のDC係数と63個のAC係数から構成されているように取り扱うことができる。

【0071】

ステップS17においては、量子化された係数は、さらに可変長符号化 (variable length coding; VLC) される。

【0072】

この可変長符号化は、DCT係数の統計的な性質を利用し、出現確率の高いものに短い符号を、出現確率の低いものに長い符号を割り当てることにより統計的冗長度を削減し、効率のよい符号化を行うもので、エントロピー符号化とも呼ばれる。

【0073】

本実施の形態での可変長符号化は、モディファイド2次元ハフマン符号化と呼ばれるものである。この符号化によると、既に出現確率がわかっている情報に対して最小の平均符号長を構成することができる。

【0074】

すなわち、ステップS17においては、各ブロックBLごとに出現する係数の出現確率から、出現頻度の高いものには少ないビット数（短い符号）を割り当て、出現頻度の低いものには多いビット数（長い符号）を割り当てることにより、各ブロックBLに割り当てるビット数を平均として少なくする。

【0075】

具体的には、図7中のAの静止モードの場合、同図中のBの動きモードの場合に示すように、量子化後の各係数をジグザクスキャン操作により一次元化した数列に対して、係数0が連続する個数であるランレングス (run length; run) とそれに続く非零係数の値 (amplitude: 振幅; amp) の対 (run, amp) に対して符号が割り当てられる。

【0076】

図8及び図9に示す可変長符号の符号割り当て表においては、符号語は、ジグザクスキャンにより得られた数列中にある非零の係数の絶対値 amp と、amp の前にある0の係数が連続する個数である run の対 (run, amp) に与え

られる。

【0077】

これは、係数を量子化してジグザグスキャンすることにより得られた数列が0を多く含む傾向があることを利用したものであり、対(run, amp)に対して出現確率に応じて可変長符号化することにより高い圧縮効率が得られる。

【0078】

図10には、対(run, amp)と符号長の関係が示されている。この図は、図8及び図9に示した可変長符号の符号割り当て表における符号割り当てを、対(run, amp)について表したものである。

【0079】

各符号語は、サインビットを含めて16ビットを超えないように設定されている。この表において、符号語の割り当てられていない対は、次式により2つの対に分割される。

$$(run, amp) = (run-1, 0) + (0, amp)$$

これにより、分割された各対は、必ず符号語が割り当てられるようになる。

【0080】

なお、図10中の記号○は、本実施の形態に基づいて置換による圧縮が可能な符号語を示すものである。この符号語置換については、さらに後述する。

【0081】

ステップS17における符号化によって得られた符号語は、図11に示すように、圧縮されたマクロブロックMBに割り当てられる。この圧縮されたマクロブロックMBは、図3に示したビデオセグメントVSを構成している。

【0082】

この圧縮されたマクロブロックMBに対するビット割り当てのフォーマットにおいては、バイト位置番号5において、上位4ビットに圧縮されたマクロブロックのエラー情報を格納するSTA (status of the compressed macroblock) が、下位4ビットに量子化器番号(QNO)が割り当てられている。

【0083】

また、バイト位置番号6からバイト位置番号19までが輝度信号Yに対応する

第0の輝度ブロックY0に、バイト位置番号20からバイト位置番号33までが
 第1の輝度ブロックY1に、バイト位置番号34からバイト位置番号47までが
 第2の輝度ブロックY2に、バイト位置番号48からバイト位置番号61までが
 第3の輝度ブロックY3にそれぞれ割り当てられている。

【0084】

さらに、バイト位置62からバイト位置71までが色差信号CRに対応するブ
 ロックBLに、バイト位置72からバイト位置81までが色差信号CBに対応す
 るブロックBLに、それぞれ割り当てられている。

【0085】

これらの各ブロックBLに割り当てられた各領域においては、最初の9ビット
 がDC係数に、これに続いて動きモードか否かを示す1ビットの動きビット(m
)及び0から3までの値を取る2ビットのクラス番号(CNO)が配置される。
 これらによって占められる、各ブロックBLにおける以降の領域は、AC係数が
 配置される。

【0086】

続いて、上述したステップS14までの処理により得られたDCT変換及び重
 み付けされたブロックBLについて行う、本実施の形態による画像圧縮の方法の
 前提となる符号化の処理を具体例を挙げて説明する。

【0087】

なお、以下では、 8×8 係数から構成されるブロックBLにおける、横に i ($0 \leq i \leq 7$) 番目、縦に j ($0 \leq j \leq 7$) 番目の係数を係数 $a_{i,j}$ として参照す
 ることにする。

【0088】

図12中のA示すブロックBLにおいては、図7に示したジグザグスキャンによ
 ると、DC係数、すなわち係数 $a_{0,0}$ が“100”、係数 $a_{1,3}$ から係数 $a_{5,0}$
 までの数列が“0, 15, 0, 0, 7”、係数 $a_{6,5}$ から係数 $a_{5,6}$ までの数列が
 “0, 9”である。なお、ジグザグスキャンにおける係数 $a_{1,3}$ の直前の係数 $a_{0,4}$
 及び係数 $a_{6,5}$ の直前の係数 $a_{7,4}$ はともに0ではないものとする。以下も同様
 に、図面に示されている0の直前の係数は0ではないものとする。

【0089】

この量子化前のブロックBLは、図5に示した量子化表に基づいて、クラス番号(CNO) 0、量子化器番号(QNO) 5にて量子化されるものとする。図6に示したエリア番号(ANO)によると、数列“0, 15, 0, 0, 7”は、エリア番号1であるので量子化ステップ2である。数列“0, 9”は、エリア番号3であるので量子化ステップ4である。

【0090】

従って、量子化を施したブロックBLは、図12中のBに示すように、数列“0, 15, 0, 0, 7”は数列“0, 7, 0, 0, 3”になる。数列“0, 9”は数列0, 2となる。

【0091】

対(run, amp)によると、数列“0, 7, 0, 0, 3”は(1, 7), (2, 3)と表される。数列“0, 2”は(1, 2)と表される。

【0092】

従って、図8及び図9に示した可変長符号化の符号化表によると、数列“0, 7, 0, 0, 3”は符号語“111010010”, 符号語“111001100”となる。数列“0, 2”は符号語“101010”となる。

【0093】

このような符号化により得られた符号語は、可変長符号とは同様な逆の復号、すなわち同じ量子化情報<QNO, CNO, ANO>により復号される。

【0094】

ここで、量子化による画像圧縮の効果について、図13を参照して説明する。図12中のAに示した量子化の前のブロックBLにおいて、図13中のAに示す、ジグザグスキャンによる数列“0, 15, 0, 0, 7”、数列“0, 9”に対して、量子化を施さずに、直接に可変長符号化する場合をまず説明することにする。

【0095】

この場合には、数列“0, 15, 0, 0, 7”は、上記対(run, amp)により、(1, 15), (2, 7)と、数列“0, 9”は(1, 9)と表される

。従って、図 8 及び図 9 に示した可変長符号化の符号化表によると、図 13 中の B に示すように、数列 “0, 15, 0, 0, 7” は符号語 “1111101111010”, 符号語 “1111101110001” と符号化される。数列 “0, 9” は符号語 “11111001000” と符号化される。すなわち、数列 “0, 15, 0, 0, 7” は 13 ビット及び 13 ビットに符号化される。数列 “0, 9” は 11 ビットに符号化される。

【0096】

一方、上述した量子化後においては、図 13 中の C に示すように、量子化した係数をジグザグスキャンした、数列 “0, 7, 0, 0, 3”、数列 “0, 2” を可変長符号化すると、図 13 中の D に示すように、それぞれ符号語 “111010010” の 9 ビット及び符号語 “111001100” の 9 ビットと、符号語 “101010” の 6 ビットに可変長符号化される。

【0097】

従って、量子化した後に可変長符号化すると、量子化をしないで可変長符号化する場合よりも、可変長符号の符号長を短くすることができる。これが、画像圧縮の際の量子化の効果である。

【0098】

すなわち、量子化により、数列 “0, 15” に対応する符号長が 13 ビットから 9 ビットに、数列 “0, 0, 7” に対応する符号長が 13 ビットから 9 ビットに、数列 “0, 9” に対応する符号長が 11 ビットから 6 ビットに低減される。

【0099】

係数の量子化とは逆に、符号化された数列からブロック B L に伸長する場合には、符号語 “111010010”, 符号語 “111001100” から (1, 7), (2, 3) が得られる。符号語 “101010” から (1, 2) が得られる。

【0100】

さらに、復元された対 (run, amp) の順序をも考慮すると、図 14 中の A に示すように、数列 “0, 7, 0, 0, 3” であり、数列 “0, 2” が得られる。また、DC 係数の値は 100 であることも同様にして明らかになっている。

【0101】

この量子化が施されたブロックBLを、量子化と同じクラス番号(CNO) 0及び同じ量子化器番号(QNO) 5にて逆量子化する。

【0102】

上述したように、数列“0, 15, 0, 0, 7”は量子化ステップ2であり、数列“0, 9”は量子化ステップ4である。このような量子化ステップによって逆量子化を施すと、図14中のBに示すように、数列“0, 14, 0, 0, 6”と、数列“8, 0”が得られる。

【0103】

図14中のBに示す、符号語から伸長されたブロックBLを、図12中のAに示す、圧縮前のブロックBLと比較すると、変化している係数がある。

【0104】

すなわち、図12中のAに示した数列“0, 15, 0, 0, 7”に対して、15は14に、7は6に変化している。また、数列“0, 9”に対して、9は8に変化している。このように、ブロックBLを符号語に圧縮して、その符号語を伸長すると、係数が元の値と変化するいわゆる劣化が見られる。

【0105】

いわゆるDV規格による量子化及び符号化による圧縮によると、このような劣化と引き換えに符号量を小さくすることができる。

【0106】

以上述べたようないわゆるDV規格による画像圧縮の前提となる符号化に基づいて、本実施の形態の画像圧縮についてさらに説明する。

【0107】

まず、符号量と量子化情報<QNO, CNO, ANO>の組み合わせによって、異なる符号語であっても同じ数列にデコードされる場合があるということについて説明する。

【0108】

例えば、図15中のAに示すように、量子化情報<5, 0, 1>、すなわち量子化器番号(QNO) 5、クラス番号(CNO) 0及びエリア番号(ANO) 1

なる条件にて可変長符号化された9ビット可変長符号語“111001100”は、図8及び図9に示した可変長符号化の符号化表により、対(run, amp)として、(2, 3)に変換され、対応する数列“0, 0, 3”が得られる。

【0109】

また、図5に示した量子化表によると、量子化情報<5, 0, 1>の量子化ステップは2である。従って、図15中のCに示すように、エリア番号(ANO)1内でジグザグスキャンした量子化前の係数の数列“0, 0, 6”に伸長される。

【0110】

一方、図15中のBに示すように、量子化情報<6, 0, 1>、すなわち量子化器番号(QNO)6、クラス番号(CNO)0及びエリア番号(ANO)1なる条件にて可変長符号化された11ビット可変長符号語“11111000110”は、図8及び図9に示した可変長符号化の符号化表により(2, 6)に変換され、図15中のCに示すように、エリア番号1内においてジグザグスキャンした数列“0, 0, 6”が得られる。また、図5に示した量子化表によると、量子化情報<6, 0, 1>の量子化ステップは1である。従って、量子化前のDCT係数の数列“0, 0, 6”に伸長される。

【0111】

このように、9ビットの符号語“111001100”及び11ビットの符号語“11111000110”に対して、異なる量子化情報<QNO, CNO, ANO>により、DCT係数の同一の数列“0, 0, 6”が得られた。このことは、適当な量子化情報<QNO, CNO, ANO>を選択することにより、係数をジグザグスキャンした数列を符号化した符号語の長さを短くすることができることを示唆している。

【0112】

本実施の形態の画像圧縮の方法は、このような符号語及び量子化情報の組み合わせによる等価性に着目したものである。すなわち、本実施の形態では、例えば量子化器番号(QNO)を変更せずに、より一般的には、量子化情報<QNO, CNO, ANO>を変更せずに、符号語単位で他の符号語に置換することによ

て、最小限の圧縮を可能にするものである。すなわち、本実施の形態においては、符号語単位での再量子化を最小限にすることにより、再量子化によるデータの劣化を最小限に止めようとするものである。

【0113】

このように量子化器番号 (QNO) 等を変更せずに符号語単位で他の符号語に置換することによる本実施の形態による画像圧縮について、具体例を挙げて説明する。

【0114】

図16中のAに示す量子化前のブロックBLにおいては、図12中のAに示した量子化前のブロックBLと同じく、ジグザクスキャンにより得られる数列として、数列“0, 15, 0, 0, 7”が示されている。

【0115】

この数列“0, 15, 0, 0, 7”を、量子化条件 $\langle \text{QNO}, \text{CNO}, \text{ANO} \rangle$ を $\langle 6, 0, 1 \rangle$ 、すなわち量子化器番号 (QNO) 6、クラス番号 (CNO) 0 及びエリア番号 (ANO) 1 により量子化することにする。

【0116】

図5に示した量子化表を参照すると、この場合の量子化ステップは1であるので、数列“0, 15, 0, 0, 7”は量子化によって変化しない。量子化された数列“0, 15, 0, 0, 7”は、対 (run, amp) により、(1, 15), (2, 7) となる。

【0117】

図8及び図9に示した可変長符号化の符号化表を参照すると、図17中のAに示すように、数列“0, 15, 0, 0, 7”に対応する13ビットの符号語“1111101111010”及び13ビットの符号語“1111101110001”が得られる。

【0118】

このように可変長符号化した符号語“1111101111010”，符号語“1111101110001”を圧縮と同一の量子化情報 $\langle 6, 0, 1 \rangle$ により伸長することにする。

【0119】

図5に示した可変長符号化表により、符号語“1111101111010”
 , 符号語“1111101110001”に対応する(1, 15), (2, 7)
 が得られる。上述したように量子化ステップは、1であるので、(1, 15),
 (2, 7)に対応する元の数列“0, 15, 0, 0, 7”に伸長される。

【0120】

このように、可変長符号化への圧縮及び伸長によって劣化が起こらないように
 する場合には、DCT係数の数列“0, 15, 0, 0, 7”は、13ビットの符
 号語及び13ビットの符号語に変換される。

【0121】

続いて、量子化器番号(QNO)を6から5に変更した場合について説明する

【0122】

図12を参照して上述したような符号化によると、図16中のAに示した数列
 “0, 15, 0, 0, 7”は、量子化情報<QNO, CNO, ANO>が<5,
 0, 1>、すなわち量子化器番号(QNO)5、クラス番号(CNO)0及びエ
 リア番号(ANO)1における量子化ステップ2により“0, 7, 0, 0, 3”
 に量子化された。

【0123】

さらに、数列“0, 7, 0, 0, 3”は、上記対(run, amp)により、
 (1, 7), (2, 3)と表され、図8及び図9に示した可変長符号化の符号化
 表を参照することにより、図17中のBに示すように、9ビットの符号語“11
 1010010”及び9ビットの符号語“111001100”に可変長符号化
 された。

【0124】

このように可変長符号化した符号語“111010010”，符号語“111
 001100”は、図16中のBに示すように、符号化とは逆の手順により、圧
 縮と同一の量子化情報<5, 0, 1>を用いることにより劣化した数列“0, 1
 4, 0, 0, 6”に伸長される。

【0125】

この方法によると、上述のように可変長符号語への圧縮と伸長によりDCT係数の劣化を発生しないように量子化器番号(QNO)6を選択した場合と比較すると、可変長符号語のビット数は13ビット及び13ビットから9ビット及び9ビットまで、すなわち8ビット削減されている。

【0126】

本実施の形態の画像圧縮の方法は、図16中のCに示すように、図16中のAに示したDCT係数の数列“0, 15, 0, 0, 7”において、数列“0, 0, 7”の部分“0, 0, 6”と置換した後に、量子化器番号(QNO)6としたまま変化させないで量子化及び可変長符号化を行うものである。

【0127】

上述したように、量子化情報<QNO, CNO, ANO>を<6, 0, 1>とすると、量子化ステップは1であるので、置換をした数列“0, 15, 0, 0, 6”は量子化によって変化しない。

【0128】

量子化した数列“0, 14, 0, 0, 6”を、上記対(run, amp)により、(1, 15), (2, 6)と表し、図8及び図9に示した可変長符号化の符号化表を参照すると、13ビットの符号語“1111101111010”及び11ビットの符号語“11111000110”が得られる。

【0129】

このように、本実施の形態の圧縮方法によると、量子化器番号(QNO)を6に保ったまま、係数をジグザグスキャンした数列を置換することにより、上述のように可変長符号語への圧縮と伸長により係数の劣化を発生しないようにした場合と比較すると、可変長符号語のビット数は13ビット及び13ビットから13ビット及び11ビットになり、すなわち2ビット削減される。

【0130】

本実施の形態の圧縮方法によると、量子化及び符号化による圧縮及び伸長によりDCT係数に劣化が起こらないように量子化条件<QNO, CNO, ANO>を選択した場合に比較して、量子化器番号(QNO)を変更したり、DCT係数

の数列を置換することによって、伸長した際にDCT係数が数列“0, 0, 7”から“0, 0, 6”と劣化することと引き換えに、圧縮した際の符号語の符号長が短縮される。

【0131】

また、本実施の形態の画像圧縮の方法によると、量子化条件 $\langle QNO, CNO, ANO \rangle$ の中で量子化器番号(QNO)を変更して圧縮した場合と比較すると、圧縮率は小さいものの、量子化器番号(QNO)が変わらないため、他のブロックBLやエリアの符号語に影響が波及しないという利点がある。

【0132】

本実施の形態の画像圧縮の方法は、図8に示した可変長符号化の符号化表の中から、量子化条件 $\langle QNO, CNO, ANO \rangle$ の中で量子化器番号(QNO)を変更して一段粗く量子化した場合と、同じデコード結果が得られるような符号語が存在する符号語に対して、符号語を置換することにより圧縮するものである。例えば、上述の例では、係数の数列“0, 0, 7”を量子化器番号(QNO)を変化させないままで数列“0, 0, 6”に置換して圧縮した。

【0133】

本実施の形態の画像圧縮の方法では、これ以外にも、 $amp = 1$ の符号語について、一段粗い量子化により $amp = 0$ の符号語に置換することにより符号長を短くすることができる場合がある。

【0134】

図10に示した可変長符号化の符号語長の表においては、本実施の形態の画像圧縮の方法による置換が可能な符号語の対(run, amp)を○にて示されている。すなわち、図に示すように、置換による符号語の長さが削減可能な符号語は、(0, 3), (1, 3), (2, 3), (3, 3), (4, 3), (5, 3), (6, 3), (0, 5), (1, 5), (4, 5), (0, 7), (2, 7), (4, 7), (0, 9), (1, 9), (1, 15)及び(0, 23)である。

【0135】

なお、上述した置換により符号語の長さの削減が可能な符号語は、例えば次の

図 18 で示すような関数を用いて求められる。

【0136】

以下、本実施の形態の圧縮方法による符号語置換による削減されるビット数を表す関数のプログラムについて、図 18 を参照して説明する。

【0137】

削減ビット数を返す関数 `sHuffmanReducible` には、削減ビット数を調べる符号語 (`nRun`, `nAmp`) を引数とする。戻り値は、削減ビット数であるが、削減されない場合には 0 を返す。

【0138】

但し、`nAmp` の範囲は、

`nAmp > 1`

または

`nAmp < -1`

である。なお、ここで除かれた `nAmp` の範囲の削減ビット数を返す関数については、後に述べる。

【0139】

この関数においては、まず `nAmp` の絶対値が取られ、絶対値が 3 より小さいか 2 より大きい場合には、削減されないとして 0 が返される。そして、符号化した符号長を返す関数 `sHuffmannLength` を用いて、`nAmp` の符号長と、`nAmp-1` の符号長の差が取られる。差が 0 より大きいときには削減ビット数としてその値を返し、そうでないときには削減されないとして 0 を返す。

【0140】

符号長を返す関数 `sHuffmannLength` は、例えば図 10 に示した符号化の符号語長の表を参照することにより実現される。

【0141】

関数 `sHuffmanReducible` の対象から除いた、

`nAmp = -1, 0, 1`

については、関数 `sHuffmanReducibleToAmp0` により扱われる。戻り値は削減ビット数であるが、削減できない場合は 0 又は負数を返す。

【0 1 4 2】

この関数は、削減ビット数を調べる符号語 (nRun,nAmp) 及び次の符号語 (nNextRun,nNextAmp) を引数とする。まず、次の符号語のnNextRunがEOBであるときには、関数sHuffmanLengthにを用いて、削減ビット数を調べる符号語のハフマン符号長を返す。

【0 1 4 3】

そうでないときには、関数sHuffmanLengthを用いて、削減ビット数を求める符号語及び削減ビット数を求める符号語の次の符号語の符号語長の和から (nNextRun,nRun+1+nNextRun) によって表される符号の符号長の差を求め、この値を削減ビット数として返す。

【0 1 4 4】

本実施の形態の圧縮方式においては、符号語のビット数を削減することが可能か否かは、量子化条件<QNO, CNO, ANO>、すなわち、量子化器番号 (QNO)、クラス番号 (CNO) 及びエリア番号 (ANO) の組に依存している。

【0 1 4 5】

量子化条件<QNO, CNO, ANO>に基づいた符号語のビット数の削減が可能であるか否かの判別は、図 1 9 に示すように、関数sQStepRducibleとして実現される。

【0 1 4 6】

この関数は、量子化条件<nQno,nClassNo,nAreaNo>、すなわち、量子化器番号nQno、クラス番号nClassNo及びエリア番号nAreaNoを引数としている。そして、量子化条件と削減可能であるかを 2 値で表した配列sQStepReductionTable0, sQStepReductionTable1, sQStepReductionTable2, sQStepReductionTable3を参照して、ビット削減が可能であるか否かを 1 又は 0 の 2 値にて返す。

【0 1 4 7】

本実施の形態の圧縮方法によると、上述したような置換を、量子化されたブロックBLの高域にある符号語から低域にある符号語に順次に適用している。図 1 6 中のCを参照した本実施の形態は、この一例である。

【0148】

すなわち、ある量子化情報 $\langle QNO, CNO, ANO \rangle$ に対して関数`sQStepReducible`により可変長符号化によるビット削減が可能であるか否かが判別される。また対 (run, amp) によって表される符号語に対して関数`sHuffmanReducible`又は`sHuffmanReducibleToAmp0`によりビット削減が可能であるか否かが判別される。

【0149】

そして、これらの関数を用いて、例えば図20に示す関数`RequantizeRunAmp`を用いて、符号語置換を行っていく。この関数は、あるブロックBLの (run, amp) 対の配数列と目標削減ビット数、量子化器番号(QNO)、クラス番号(CNO)を入力し、置換後の結果を対 (run, amp) による配列に格納するものである。

【0150】

すなわち、この関数`RequantizeRunAmp`は、 (run, amp) の配列(`pRuns`, `pAmps`)、このブロックBLの目標削減ビット数`nReducingSize`、このブロックBLの量子化器番号`nQno`及びクラス番号`nClassNo`を引数としている。

【0151】

この関数`RequantizeRunAmp`においては、以下のような手順により処理が行われる。まず、図20中のaの部分においては、入力された対 $(pRuns, pAmps)$ の配列に対するエリア分けが施される。すなわち、入力された $(pRuns, pAmps)$ の配列は、図6に示したようなエリアに分けられる。

【0152】

図21中のbの部分においては、図19で示した量子化条件 $\langle QNO, CNO, ANO \rangle$ によりビット削減が可能であるか否かを判別する関数`sQStepReducible`を用いて、エリア分けされた各領域が削減可能であるかを調べる。

【0153】

図21中のcの部分においては、関数`sQStepReducible`により削減可能であると判断されたときには、図18に示した関数`sHuffmanReducible`又は`sHuffmanReducibleToAmp0`を用いて、各エリア内の配列(`pRuns, pAmps`)を後ろ、すなわち高

域から削減可能であるかを調べ、削減可能であるときにはビット数削減するように符号語を置換する。

【0 1 5 4】

図 2 1 中の d の部分においては、そのエリアについて、符号語を一つ置換する毎に、これまでの削減ビット数の合計を調べる。そして、目標削減ビット数に達していれば終了し、削減したビット数の総数を返す。

【0 1 5 5】

図 2 1 中の e の部分においては、配列 (pRuns, pAmps) について全てのエリアについて上述の処理が行われたので、最後に削減されたビット数を返す。

【0 1 5 6】

この関数 RequantizeRunAmp では、対 (run, amp) に対して置換処理を行ったが、後述する図 2 3 に示すような入力コード (input code) と新コード (new code) との対応関係の表を用いて、可変長符号化されて符号語とされた DCT 係数を直接に置換することも可能である。

【0 1 5 7】

例えば、置換可能なエリアに 1 0 0 0 0 という符号語が見つかった場合、図 2 2 の対応関係の表を参照することにより、これを 0 1 0 0 に置換することができる。

【0 1 5 8】

ブロック B L の係数をジグザクスキャンした数列において高域から低域にビット削減可能な符号語を順次に削減していく過程において、ターゲットビット数、すなわち目標符号量に達した場合、それ以上置換する必要はなく、低域部は元の値のままとなる。一般に低域部には画像情報の重要な係数が集中しており、低域部が維持されれば画質劣化が少ないといえる。

【0 1 5 9】

ブロック B L 内すべての符号語に対して、上記の置換を施しても目標符号量が得られない場合、図 1 6 中の B に示した方法と同様に、量子化器番号 (QNO) を変更する。それでも目標符号量に達しない場合、新しい量子化器番号に対して本実施の形態による圧縮方式による置換を行う。これを目標符号量に達するまで

繰り返す。

【0 1 6 0】

なお、各ブロック B L の目標符号量の決め方の一例として、図 2 3 に示すように、ビデオセグメント V S 容量を越えたビット数を各ブロック B L に分配し、これを各ブロック B L の削減目標符号量 R_{mn} とする。各ブロック B L に対して、本実施の形態による符号語単位での圧縮を行い、削減量の合計が R_{mn} に達するまで反復する。

【0 1 6 1】

ここで、圧縮前の各ブロック B L の符号量を B_{mn} ビットとする。ここで、 m ($0 < m < 4$) はマクロブロック M L の番号であり、 n ($0 < n < 5$) はブロック B L の番号である。

【0 1 6 2】

ビデオセグメントの容量を M ビットとするとき、ビデオセグメント V S 全体で削減しなければならないビット数 R は、

$$R = \sum_{i=0}^4 \sum_{j=0}^5 B_{ij} - M$$

である。これより、各ブロック B L の削減目標 R_{mn} は、式

$$R_{mn} = R \times B_{mn} / \sum_{i=0}^4 \sum_{j=0}^5 B_{ij}$$

により求められる。

【0 1 6 3】

また、以上説明した本実施の形態による画像圧縮の方式は、原画像を圧縮する場合にも適用することができる。

【0 1 6 4】

従来は、目標符号量が得られるまでブロック B L 単位で、ターゲットビットを目標として何度か圧縮し直していた。すなわち、再量子化に続いて再可変長符号化を行うことを繰り返していた。

【0 1 6 5】

しかし、本実施の形態の画像圧縮の方法によると、一度圧縮した結果を元にさらに「符号語単位」で圧縮することができるため、より目標符号量に近い符号化を行うことができる。

【0 1 6 6】

すなわち、本実施の形態の画像圧縮の方法によると、決められた符号容量をより効率よく利用でき、画質劣化がそれだけ少なくて済む。さらに 2 回目以降の圧縮が符号語の置換で行うため、ブロック B L 全体を圧縮し直す従来法より高速に処理できる。

【0 1 6 7】

さらに、本実施の形態による画像圧縮の方法は、別の応用として、既に圧縮された画像の加工の際にも適用できる。

【0 1 6 8】

従来、圧縮画像に対してタイトル挿入や画面合成といった合成処理を行う場合など、一度フレーム全体を伸長し直した後、所望の処理を施し、再圧縮するのが通例であった。

【0 1 6 9】

これに対し、より効率的で、演算誤差に伴う劣化を回避する方法として、必要最低限の部分のみ伸長 際圧縮する方法が提案されている。この方法はデジタル信号を、加工処理単位としてのブロック B L に分割し、このブロック B L ごとに独立に上記デジタル信号を加工処理するものである。これにより、例えば画像全体において編集が必要な最小限の部分のみについて加工処理を行うことができる。

【0 1 7 0】

~~この方法によれば、合成処理されるまでの部分を含む周辺のみ伸長・合成処理~~
・再圧縮が必要となり、フレーム内の残りの大部分は圧縮されたまま複製すれば済む。

【0 1 7 1】

しかし、一般に圧縮画像はある単位ごとの符号量に上限が決められており、複製しただけではこの上限値を超えてしまう場合がある。この場合、何らかの再圧縮が必要になる。このようなとき、本実施の形態による圧縮方式により効率よく、劣化を少なく再圧縮することが可能である。

【0 1 7 2】

二つの入力DV画像A、Bを合成して出力DV画像Cを得る場合について、図24を参照して説明する。

【0 1 7 3】

このとき、合成境界にかからないブロックBLは圧縮されたまま複製可能である。また、合成境界にかかるブロックBLは例えば伸長し、圧縮し、再圧縮することで圧縮された合成ブロックBLを生成することができる。

【0 1 7 4】

いま、セグメントの中の5マクロブロックMBの内、4つがAからの複製で、1つがBからの複製、又は合成ブロックBLであるとする。このとき、Bから複製ブロックBL、又は合成ブロックBLの符号量によってはビデオセグメントVS容量を越えてしまい、再圧縮する必要がある場合がある。

【0 1 7 5】

このような場合に、本実施の形態による圧縮方法により、可変長符号語レベルで圧縮することが効果的である。

【0 1 7 6】

なお、上述の実施の形態においては、いわゆるDV形式のフォーマットに対して適用されるものとして説明したが、本実施の形態がこれに限定されるものでないことはもちろんである。

【0 1 7 7】

【発明の効果】

上述のように、本発明によると、可変長符号語に対して、量子化番号等の量子化情報を変更することなく、さらに圧縮することができる。

【0 1 7 8】

また、本発明によると、符号語単位で圧縮できるため、例えばDCT係数の低域は変更せず、高域のみ本方式により部分的に圧縮することが出来、より劣化の少ない、すなわち元の信号をより多く維持した画像が得られる。このため、最悪でも量子化番号等を変更する従来の圧縮と同じ結果を得ることができる。

【0179】

さらに、本発明によると、符号語という小さな単位で処理するため、より目標符号量に近づけることができ、符号容量を有効利用できる。

【0180】

そして、本発明によると、符号語に対して、デコードせずに置換するだけなので、処理が簡単で高速に行える。また、本発明によると、エンコード時などに行った符号化が無駄になりにくい。

【0181】

さらにまた、本発明によると、既に圧縮された画像に対して、一部変更した結果、圧縮が必要になった場合に、再量子化及び再符号化せずにさらに圧縮できる。

【図面の簡単な説明】

【図1】

本発明の実施の形態を適用した編集装置の概略的な構成を示すブロック図である。

【図2】

編集装置の画像編集ソフトウェアに関連する部分の構成を示すブロック図である。

【図3】

いわゆるDV規格によるフォーマットを示す図である。

【図4】

いわゆるDV規格により圧縮画像を作成する手順を示すフローチャートである。

【図5】

量子化器番号、クラス番号及びエリア番号による量子化ステップを示す量子化表を示す図である。

【図6】

ブロックのエリア分割を示す図である。

【図 7】

ブロックにおけるジグザグスキャンを示す図である。

【図 8】

対 (run, amp) からの可変長符号語への変換を示す図である。

【図 9】

対 (run, amp) からの可変長符号語への変換を示す、図 8 に続く図である。

【図 1 0】

可変長符号語と符号長との関係を示す図である。

【図 1 1】

圧縮されたマクロブロックに対するビット割り当てを示す図である。

【図 1 2】

量子化及び可変長符号化による圧縮の例を示す図である。

【図 1 3】

量子化による圧縮の効果を説明する図である。

【図 1 4】

量子化及び可変長符号化により圧縮された符号語からの伸長の例を示す図である。

【図 1 5】

異なった量子化器番号によって同一の伸長結果が得られる例を示す図である。

【図 1 6】

本実施の形態による圧縮を示す図である。

【図 1 7】

本実施の形態による圧縮の効果を説明する図である。

【図 1 8】

符号語置換による削減ビット数を返す関数のプログラムを示す図である。

【図 1 9】

量子化情報により削減可能か否かを判定する関数のプログラムを示す図である。

【図 2 0】

符号語置換を行う関数のプログラムを示す図である。

【図 2 1】

符号語置換を行う関数のプログラムを示す図である。

【図 2 2】

符号語を置換する表を示す図である。

【図 2 3】

各ブロックの削減目標ビット数の求め方を説明する図である。

【図 2 4】

ワイプの処理を示す図である。

【図 2 5】

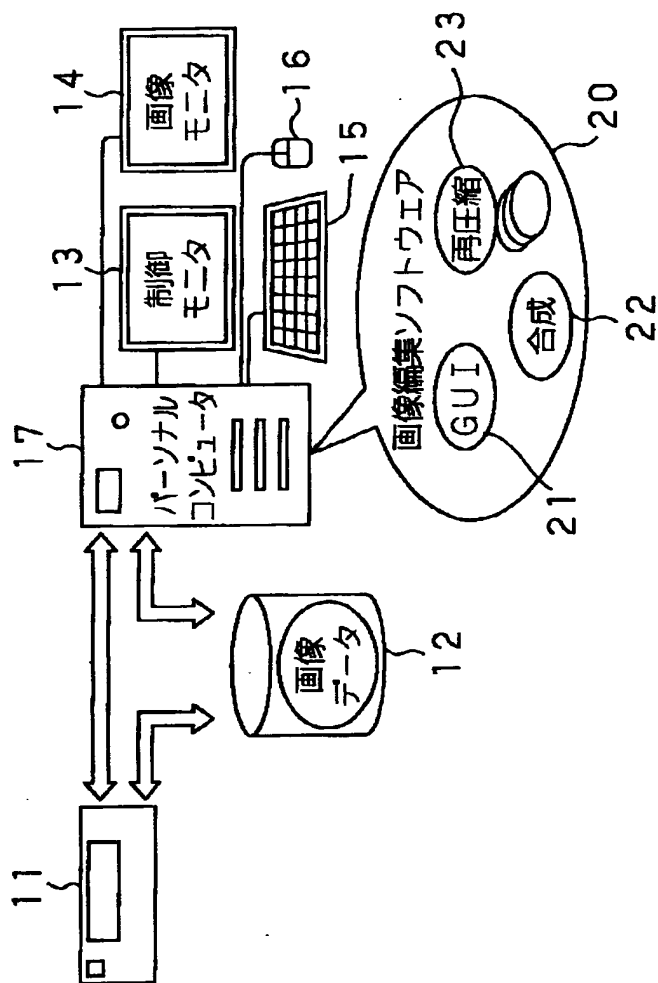
デジタルビデオの編集加工処理を示す図である。

【符号の説明】

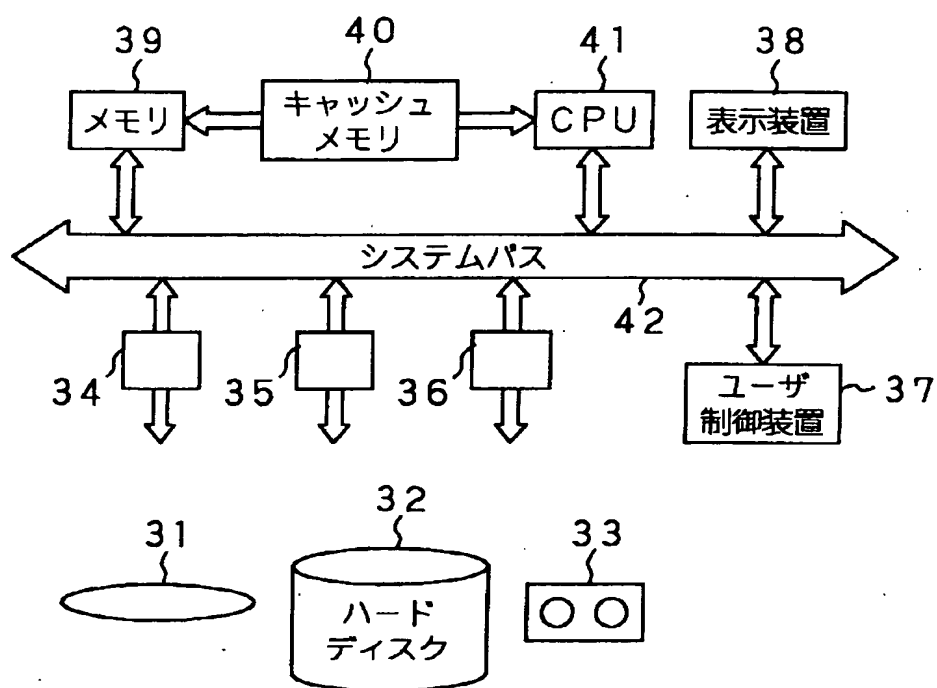
1 1 カメラデッキ、1 2 ハードディスクドライブ、1 3 制御モニタ、1 4 画像モニタ、1 7 パーソナルコンピュータ、2 0 画像編集ソフトウェア、2 1 グラフィックユーザインターフェース、2 2 合成処理プログラム、2 3 再圧縮処理プログラム、3 9 メモリ、4 1 CPU、3 7 ユーザ制御装置、3 8 表示装置

【書類名】 図面

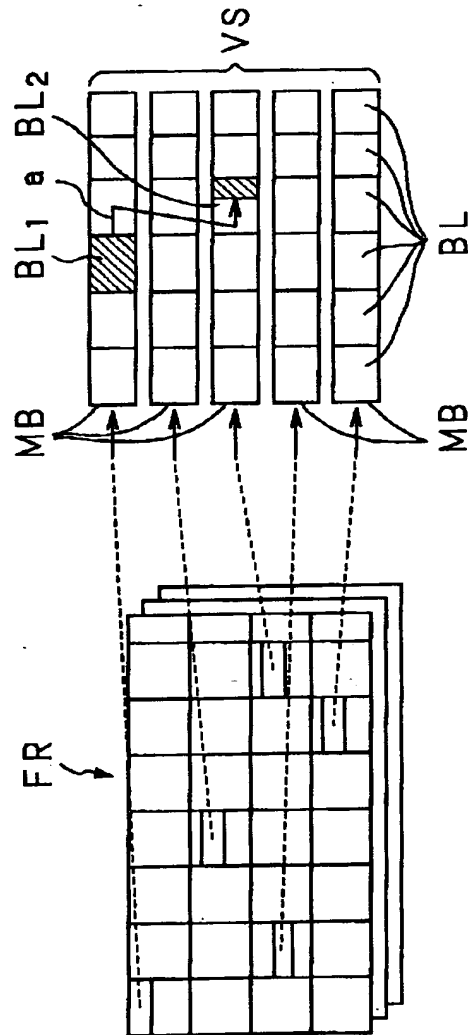
【図 1】



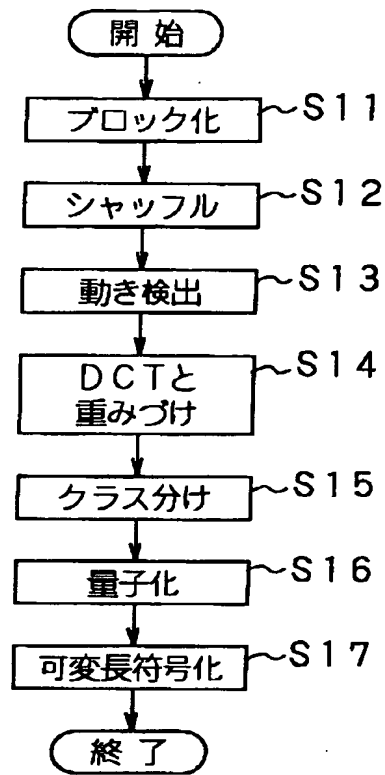
【図 2】



【図 3】



【図 4】



【図 5】

	クラス番号 (CNO)				エリア番号 (ANO)			
	0	1	2	3	0	1	2	3
量子化器番号 (QNO)	15				1	1	1	1
	14				1	1	1	1
	13				1	1	1	1
	12	15			1	1	1	1
	11	14			1	1	1	1
	10	13		15	1	1	1	1
	9	12	15	14	1	1	1	1
	8	11	14	13	1	1	1	2
	7	10	13	12	1	1	2	2
	6	9	12	11	1	1	2	2
	5	8	11	10	1	2	2	4
	4	7	10	9	1	2	2	4
	3	6	9	8	2	2	4	4
	2	5	8	7	2	2	4	4
	1	4	7	6	2	4	4	8
	0	3	6	5	2	4	4	8
		2	5	4	4	4	8	8
		1	4	3	4	4	8	8
		0	3	2	4	8	8	16
			2	1	4	8	8	16
			1	0	8	8	16	16
			0		8	8	16	16

【図 6】

A 8-8-DCT

		水平 →							
		0	1	2	3	4	5	6	7
垂直 ↓	0	DC	0	0	1	1	1	2	2
	1	0	0	1	1	1	2	2	2
	2	0	1	1	1	2	2	2	3
	3	1	1	1	2	2	2	3	3
	4	1	1	2	2	2	3	3	3
	5	1	2	2	2	3	3	3	3
	6	2	2	2	3	3	3	3	3
	7	2	2	3	3	3	3	3	3

B 2-4-8-DCT

		水平 →							
		0	1	2	3	4	5	6	7
(和) 垂直 ↓ (差)	0	DC	0	1	1	1	2	2	3
	1	0	1	1	2	2	2	3	3
	2	1	1	2	2	2	3	3	3
	3	1	2	2	2	3	3	3	3
	4	0	0	1	1	2	2	2	3
	5	0	1	1	2	2	2	3	3
	6	1	1	2	2	2	3	3	3
	7	1	2	2	3	3	3	3	3

【図 7】

A 8-8-DCT

		水平 →							
		0	1	2	3	4	5	6	7
0		1	2	6	7	15	16	28	29
1		3	5	8	14	17	27	30	43
2		4	9	13	18	26	31	42	44
3	垂直 ↓	10	12	19	25	32	41	45	54
4		11	20	24	33	40	46	53	55
5		21	23	34	39	47	52	56	61
6		22	35	38	48	51	57	60	62
7		36	37	49	50	58	59	63	64

B 2-4-8-DCT

		水平 →							
		0	1	2	3	4	5	6	7
0		1	3	7	19	21	35	37	51
1	(和)	5	9	17	23	33	39	49	53
2		11	15	25	31	41	47	55	61
3	垂直 ↓	13	27	29	43	45	57	59	63
4	(差)	2	4	8	20	22	36	38	52
5		6	10	18	24	34	40	50	54
6		12	16	26	32	42	48	56	62
7		14	28	30	44	46	58	60	64

【図 8】

(run,amp)	コード	長さ	(run,amp)	コード	長さ
0 1	00s	2+1	11 1	111100000s	9+1
0 2	010s	3+1	12 1	111100001s	
EOB	0110	4	13 1	111100010s	
1 1	0111s	4+1	14 1	111100011s	
0 3	1000s		5 2	111100100s	
0 4	1001s		6 2	111100101s	
2 1	10100s	5+1	3 3	111100110s	
1 2	10101s		4 3	111100111s	
0 5	10110s		2 4	111101000s	
0 6	10111s		2 5	111101001s	
3 1	110000s	6+1	1 8	111101010s	10+1
4 1	110001s		0 18	111101011s	
0 7	110010s		0 19	111101100s	
0 8	110011s		0 20	111101101s	
5 1	1101000s	7+1	0 21	111101110s	
6 1	1101001s		0 22	111101111s	
2 2	1101010s		5 3	1111100000s	
1 3	1101011s		3 4	1111100001s	
1 4	1101100s		3 5	1111100010s	
0 9	1101101s		2 6	1111100011s	
0 10	1101110s		1 9	1111100100s	
0 11	1101111s	8+1	1 10	1111100101s	
7 1	11100000s		1 11	1111100110s	
8 1	11100001s		0 0	11111001110s	11
9 1	11100010s		1 0	11111001111s	
10 1	11100011s		6 3	11111010000s	11+1
3 2	11100100s		4 4	11111010001s	
4 2	11100101s		3 6	11111010010s	
2 3	11100110s		1 12	11111010011s	
1 5	11100111s		1 13	11111010100s	
1 6	11101000s	12	1 14	11111010101s	12
1 7	11101001s		2 0	111110101100	
0 12	11101010s		3 0	111110101101	
0 13	11101011s		4 0	111110101110	
0 14	11101100s		5 0	111110101111	
0 15	11101101s				
0 16	11101110s				
0 17	11101111s				

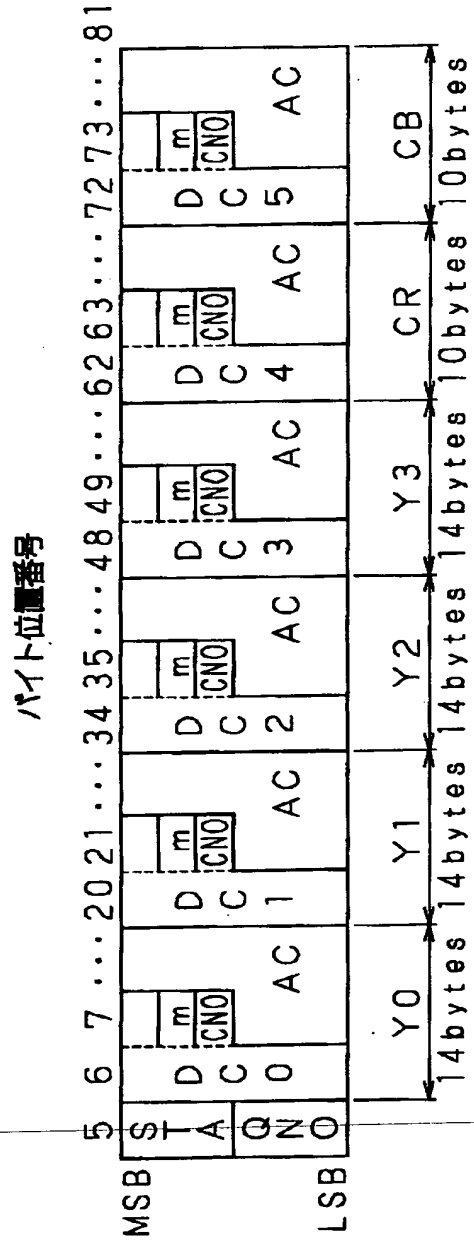
【図9】

(run, amp)		コード		長さ
7	2	111110110000	s	12+1
8	2	111110110001	s	
9	2	111110110010	s	
10	2	111110110011	s	
7	3	111110110100	s	
8	3	111110110101	s	
4	5	111110110110	s	
3	7	111110110111	s	
2	7	111110111000	s	
2	8	111110111001	s	
2	9	111110111010	s	
2	10	111110111011	s	
2	11	111110111100	s	
1	15	111110111101	s	
1	16	111110111110	s	
1	17	111110111111	s	
6	0	1111110000110		13
7	0	1111110000111		
⋮	⋮			
R	0	1111110	Rの2進表記 R=6~61	
⋮	⋮			15+1
61	0	1111110111101		
0	23	111111100010111	s	
0	24	111111111111000	s	
⋮	⋮			
0	A	1111111	Aの2進表記 A=23~255	s
⋮	⋮			
0	255	111111111111111	s	

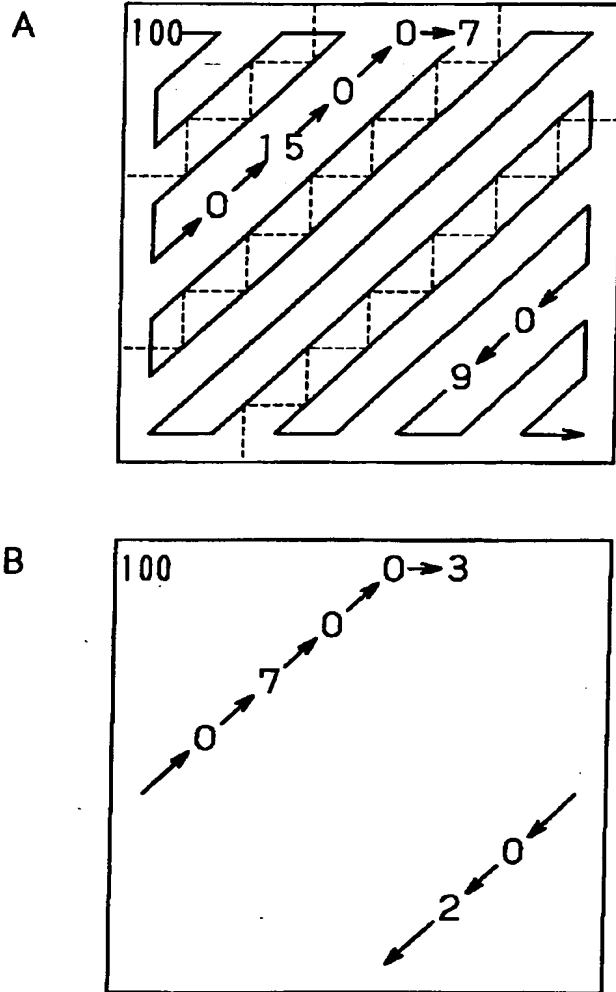
【図 1 0】

		amp																								
run	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	11	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	11	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
2	12	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
3	12	7	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
4	12	7	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
5	12	8	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
6	13	8	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
7	13	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
8	13	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
9	13	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
10	13	9	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
11	13	10																								
12	13	10																								
13	13	10																								
14	13	10																								
15	13																									
:	:																									
61	13																									

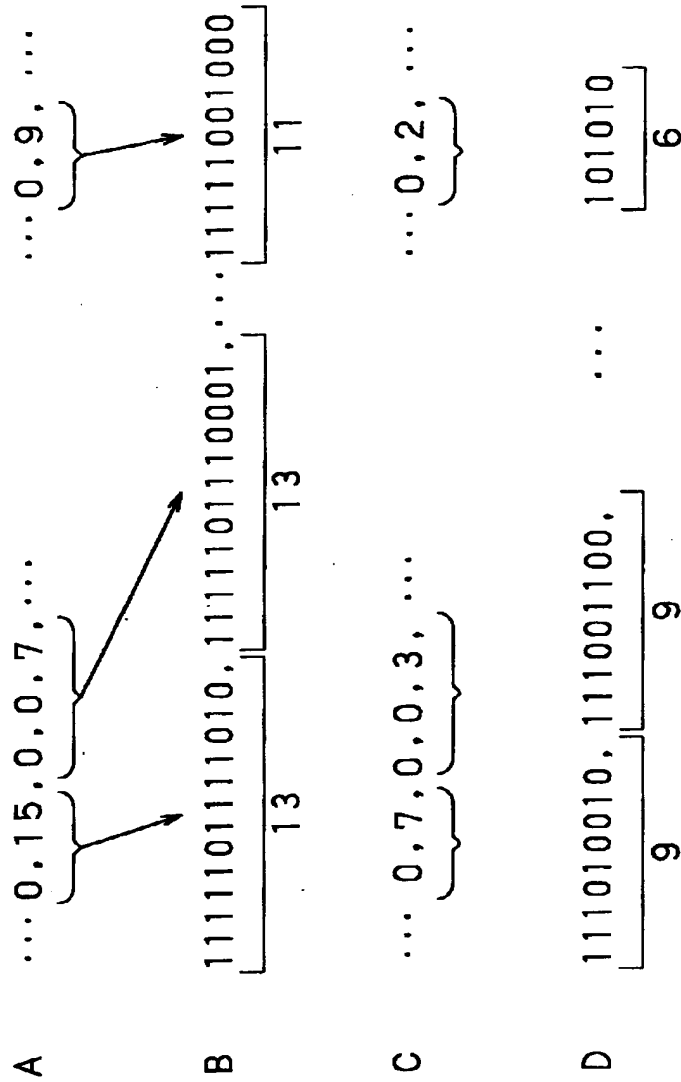
【図 1 1】



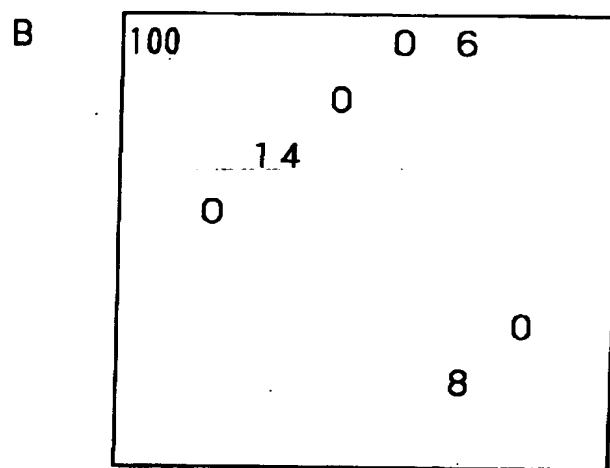
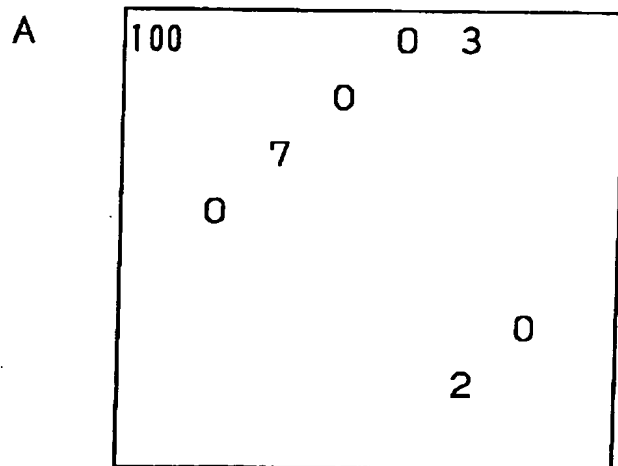
【図 1 2】



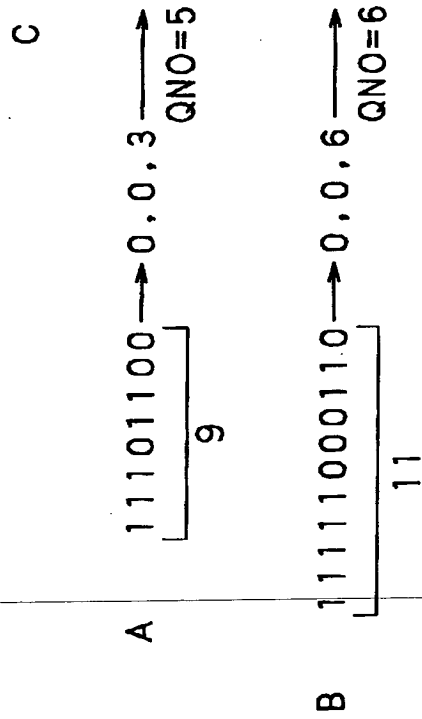
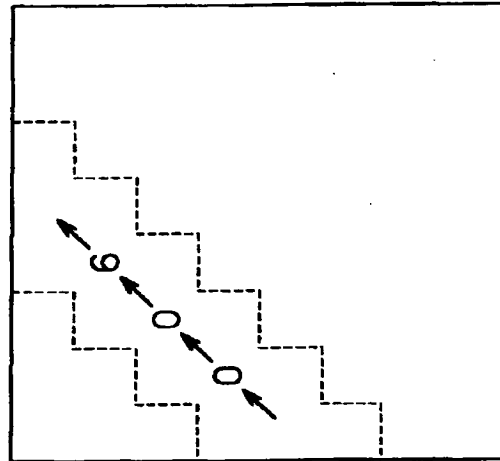
【図 1 3】



【図 1 4】

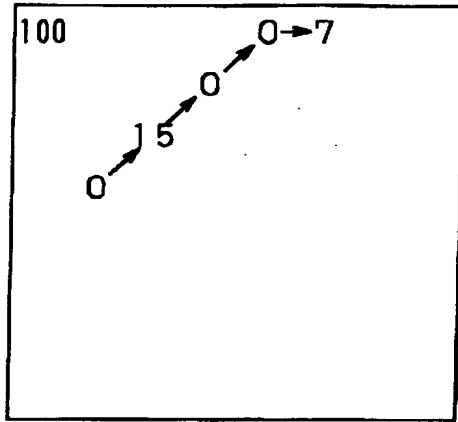


【図 1 5】

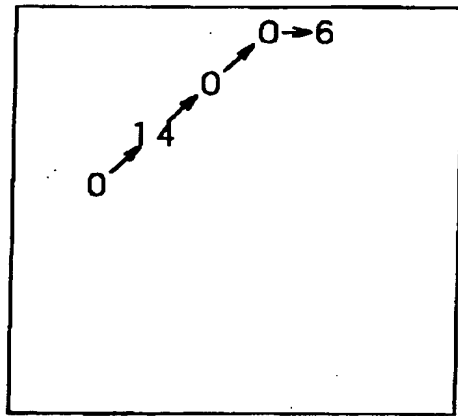


【図 1 6】

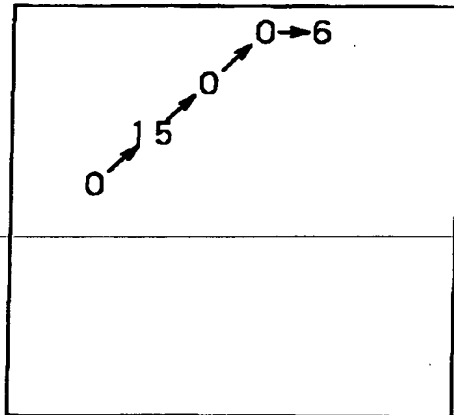
A



B



C



【図 1 7】

A	QNO=6 →	1111101111010,	0.15	(1, 15)
		1111101110001,	0.17	(2, 7)
B	QNO=5 →	111010010,	0.7	(1, 7)
		111001100,	0.0, 3	(2, 3)
C	QNO=6 →	1111101111010,	0.15	(1, 15)
		11111000110,	0.0, 6	(2, 6)

【図 1 8】

```

static int sHuffmanReducible(int nAmp, int nRun)
{
    if(nAmp<0)
        nAmp=-nAmp;
    if((nAmp<3)|| (nAmp>23))
        return 0;
    int nThisLength=sHuffmanLength(nAmp, nRun);
    int nNewLength=sHuffmanLength(nAmp-1, nRun);
    int nReduced=nThisLength-nNewLength;
    return(nReduced>0)?nReduced:0;
}

static int sHuffmanReducibleToAmp0(int nAmp, int nRun, int nNextAmp, int nNextRun)
{
    int nThisLength=sHuffmanLength(nAmp, nRun);
    if(nNextRun==EOB){/*the end of data*/
        return nThisLength;
    }
    int nNextLength=sHuffmanLength(nNextAmp, nNextRun);
    int nNewLength=sHuffmanLength(nNextAmp, nRun+1+nNextRun);
    return nThisLength+nNextLength-nNewLength;
}

```

【図 1 9】

```

static BOOL s0StepReductionTable0[4][16] = {
    {0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}
};

static BOOL s0StepReductionTable1[4][16] = {
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0}
};

static BOOL s0StepReductionTable2[4][16] = {
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0},
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0}
};

static BOOL s0StepReductionTable3[4][16] = {
    {0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1},
    {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0}
};

static BOOL s0StepReducible(int n0no, int nClassNo, int nAreaNo)
{
    switch (nAreaNo) {
        case (3):
            return s0StepReductionTable3[nClassNo][n0no];
        case (2):
            return s0StepReductionTable2[nClassNo][n0no];
        case (1):
            return s0StepReductionTable1[nClassNo][n0no];
        case (0):
            return s0StepReductionTable0[nClassNo][n0no];
        default:
            return FALSE;
    }
}

```


【図 20】

```

#define EOB 62
int RequantizeRunAmp(short* pRuns, short* pAmps, int nReducingSize, int nOno, int nClassNo)
{
    static int area[4] = { 1, 6, 21, 43 };

    int nAreaEnd[4];

    for (int i = 0; i < 4; i++) {
        nAreaEnd[i] = -1;
    }

    int nArea = 1;
    int nCoef = 0;

    for (i = 0; pRuns[i] != EOB; i++) {
        if (pRuns[i] > EOB) // skip invalid entry
            continue;

        nCoef += pRuns[i] + 1;

        if ((nArea < 4) && (nCoef >= area[nArea])) {
            nAreaEnd[nArea-1] = i-1;
            nArea++;
        }
    }

    int nLastEntry = i - 1;

    for (i = 0; i < 4; i++) {
        if (nAreaEnd[i] == -1) {
            nAreaEnd[i] = nLastEntry;
        }
    }

    int nTotalReducedSize = 0;

    int nReducedSize;
    int nLastAmp;
    int nLastRun;
    int nLastPos;
}

```

【図 2 1】

```

for (i = 3; i >= 0; i--) {
    if (s0StepReducible(nOno, nClassNo, i)) ~b

        if (nAreaEnd[i] < nLastEntry) {
            nLastPos = nAreaEnd[i]+1;
            nLastRun = pRuns[nLastPos];
            nLastAmp = pAmps[nLastPos];
        } else {
            nLastPos = nLastEntry;
            nLastRun = EOB;
        }

        int nPrevAreaEnd = (i>0)?nAreaEnd[i-1]:-1;

        /* 現在のエリア内で高域のエントリから順次置換する */
        for (int j = nAreaEnd[i]; j > nPrevAreaEnd; j--) {
            if (pRuns[j] >= EOB)
                continue;

            /* ampが1または-1または0の場合 */
            if ((pAmps[j] <= 1) || (pAmps[j] >= -1)) {
                nReducedSize = shuffmanReducibleToAmp0(pAmps[j], pRuns[j], nLastAmp, nLastRun);
                if (nReducedSize >= 0) {
                    if (nLastRun == EOB) {
                        pRuns[j] = EOB; // set the end of data
                    } else {
                        pAmps[j] = nLastAmp;
                        pRuns[j] += nLastRun + 1; // runを加算
                        pRuns[nLastPos] = 255; // エントリが無効であることをマーク
                    }
                }
            } else {
                /* ampがその他の場合 */
                nReducedSize = shuffmanReducible(pAmps[j], pRuns[j]);
                if (nReducedSize > 0)
                    pAmps[j] = (pAmps[j]>0)?1:(-1); /* ampを置換 */
            }

            if (nReducedSize > 0) {
                nTotalReducedSize += nReducedSize;
                if (nTotalReducedSize >= nReducingSize)
                    return nTotalReducedSize; /* 目標削減量に達したので終了 */ ~d
            }

            nLastAmp = pAmps[j];
            nLastRun = pRuns[j];
            nLastPos = j;
        }
    }

    return nTotalReducedSize; ~e
}

```

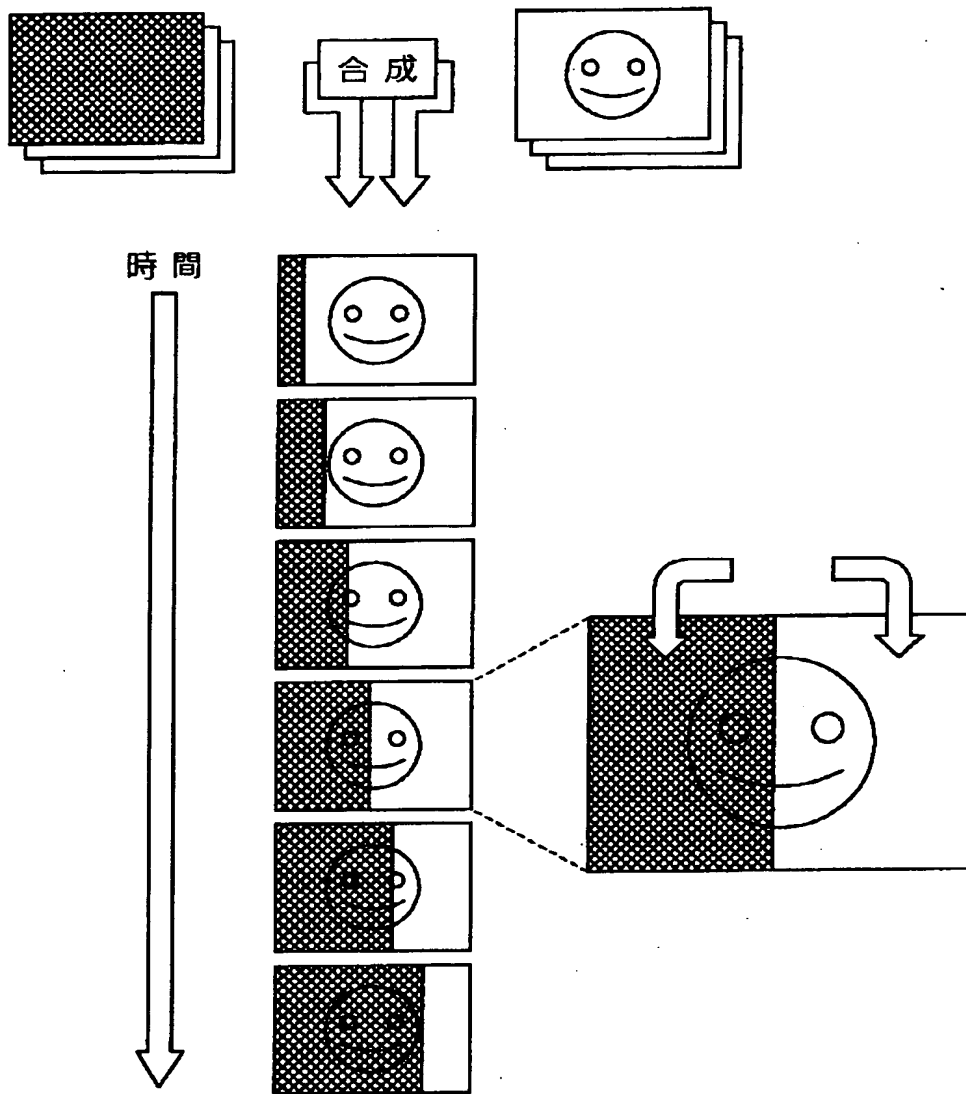
【図 22】

入力符号語	新符号語	入力(run, amp)→新(run, amp)
1000s	010s	(0,3)→(0,2)
10110s	1001s	(0,5)→(0,4)
110010s	10111s	(0,7)→(0,6)
1101101s	110011s	(0,9)→(0,8)
11111100010111s	111101111s	(0,23)→(0,22)
1101011s	10101s	(1,3)→(1,2)
11100111s	1101100s	(1,5)→(1,4)
1111100100s	111101010s	(1,9)→(1,8)
111110111101s	11111010101s	(1,15)→(1,14)
11100110s	1101010s	(2,3)→(2,2)
111110111000s	1111100011s	(2,7)→(2,6)
111100110s	11100100s	(3,3)→(3,2)
111110110111s	11111010010s	(3,7)→(3,6)
111100111s	11100101s	(4,3)→(4,2)
111110110110s	11111010001s	(4,5)→(4,4)
1111100000s	111100100s	(5,3)→(5,2)
11111010000s	111100101s	(6,3)→(6,2)

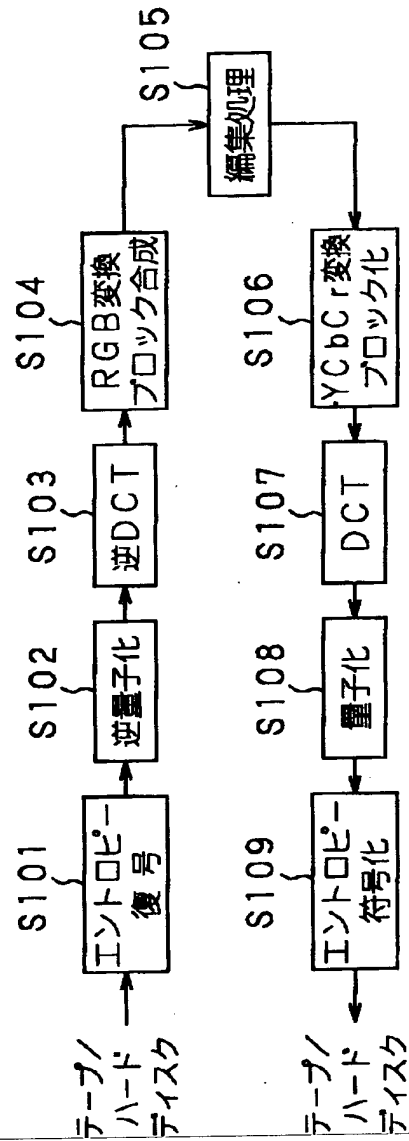
【図 2 3】

	Y ₀	Y ₁	Y ₂	Y ₃	Cr	Cb
MB0	B ₀₀	B ₀₁	B ₀₅
MB1	B ₁₀	B ₁₁				
MB2	B ₂₀	⋮	⋱			
MB3	B ₃₀	⋮		⋱		
MB4	B ₄₀	⋮				B ₄₅

【図 2 4】



【図 2 5】



【書類名】 要約書

【要約】

【課題】 可変長符号語を圧縮する。

【解決手段】 DVカメラデッキ 1 1 やハードディスクドライブ 1 2 から供給される可変長符号化によるDV規格の圧縮画像に対し、パーソナルコンピュータ 1 7 で実行される画像編集ソフトウェア 2 0 である再圧縮処理プログラム 2 3 は、上記可変長符号語の符号語列中の一部の符号語を、長さがより短くかつ値が近似した他の符号語に置換する。

【選択図】 図 1

認定・付加情報

特許出願の番号	平成11年 特許願 第193542号
受付番号	59900653441
書類名	特許願
担当官	第三担当上席 0092
作成日	平成11年 7月 9日

<認定情報・付加情報>

【提出日】	平成11年 7月 7日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社

THIS PAGE BLANK (USPTO)